AMENDMENTS TO THE CLAIMS:

1. (Currently amended) A method of executing a linear algebra subroutine on a computer having at least one cache, said method comprising:

streaming data ~~for~~ from two of three matrices involved in processing said linear algebra subroutine to a first matrix such that submatrix data ~~is processed using data for a first matrix residing in said cache as one of an entirety of said first matrix and a submatrix of said first matrix and data from a second matrix and a third matrix is respectively residing as one of its entirety and submatrices thereof in a memory device at a higher level than said cache~~ of said two matrices residing in a higher level cache or in a memory is streamed to submatrix data of said first matrix residing in said cache,

said streaming providing data from said higher level ~~as~~ to said data in said cache ~~is~~ as required for said processing for executing correctly the linear algebra subroutine.

2. (Original) The method of claim 1, wherein said at least one cache comprises an L1 cache and said higher level comprises an L2 cache.

3. (Currently amended) The method of claim 1, further comprising:

selecting said matrix of the three to be stored in said cache by ~~determining which matrix will fit into said cache~~ examining sizes and shapes of said three matrices.

4. (Currently amended) The method of claim 3, further comprising:

determining a size of each of said first matrix, said second matrix, and said third matrix;

determining which of said first matrix, said second matrix, and said third matrix will fit into a size of said cache;

arranging elements of elements of said first, second, and third matrices for said streaming; and

loading data for a selected one of said first matrix, said second matrix, and said third matrix into said cache.

5. (Currently amended) The method of claim 1, further comprising:

selecting a linear algebra subroutine from a plurality of subroutines to perform a matrix operation, said selecting based on which of said plurality of subroutine has a format consistent with said matrix stored in said cache, said plurality of subroutines comprising six subroutines each capable of performing said matrix operation using one matrix operand as being cache resident and a remaining two matrix operands as streaming through said cache from said higher level cache or memory.

6. (Currently amended) The method of claim 2, wherein data for said second matrix and said third matrix streams into said L1 cache from said L2 cache such that said data from ~~one of~~ said second matrix and said third matrix streams in a vector format into said L1 cache ~~and data from the other of said second matrix and said third matrix streams in a scalar format~~.

7. (Original) The method of claim 1, wherein said linear algebra subroutine comprises a subroutine from a LAPACK (Linear Algebra PACKage).

8. (Currently amended) The method of claim 7, wherein said ~~LAPACK~~ subroutine comprises a BLAS Level 3 routine or a BLAS Level 3 ~~L1 cache~~ kernel routine.

9. (Currently amended) An apparatus, comprising:

a memory system to store matrix data for processing in a linear algebra program using data from a first matrix, a second matrix, and a third matrix, said memory system including at least one cache; and

a processor to perform a linear algebra operation, wherein data from one of said first matrix, said second matrix, and said third matrix is stored in said cache in a matrix format and data from a remaining two matrices is stored in said memory system at a level higher that said cache,

said data from said remaining two matrices being streamed into said processor as required by said processing by streaming data from two of three matrices involved in processing said linear algebra subroutine to a first matrix such that submatrix data of said two matrices residing in a higher level cache or in a memory is streamed to submatrix data of said first matrix residing in said cache.

10. (Currently amended) The apparatus of claim 9, further comprising:

a selector to determine a size of each of matrices involved in a matrix multiplication process and to select one of said matrices to reside in said cache, as based on having determined said sizes;

a loader to load data for the selected matrix into said cache; and

a selector to select a matrix subroutine, from a plurality of matrix subroutines, to perform said ~~linear algebra program processing~~ matrix multiplication process, each matrix subroutine in said plurality capable of executing said matrix multiplication process using one matrix operand as being cache resident and a remaining two matrix operands as streaming through said cache from said higher level cache or memory,

said selected matrix subroutine having a format consistent with which said matrix is selected to reside in said cache.

11. (Original) The apparatus of claim 9, wherein said linear algebra program comprises a subroutine from a LAPACK (Linear Algebra PACKage).

12. (Currently amended) The apparatus of claim 11, wherein said ~~LAPACK~~ subroutine comprises a BLAS Level 3 routine or a BLAS Level 3 ~~L1 cache~~ kernel ~~types~~ routine.

13. (Currently amended) The apparatus of claim 10, wherein said plurality of matrix subroutines comprises ~~two of~~ six possible matrix subroutines.

14. (Original) A signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of ~~of~~ executing a linear algebra subroutine on a computer having at least one lower level cache and one or more higher level caches or other higher level memory devices, said method comprising:

streaming data for up to three matrices involved in processing said linear algebra subroutine such that data is processed using data for a first matrix stored in said cache as serving a matrix ~~format~~ role in said linear algebra subroutine and being cache resident and data from a second matrix and a third matrix is stored ~~in a memory device~~ at a higher level than said cache, said streaming providing data from said higher level in a manner as said data is required for said processing by streaming data from said second and third matrices to submatrix data of said first matrix residing in said cache.

15. (Currently amended) The signal-bearing medium of claim 14, said method further comprising:

determining a size of each of matrices involved in a matrix multiplication process;

selecting one of said matrices to reside in one of said at least one cache, based on having determined said sizes;

arranging elements of elements of said matrices for said streaming; and

selecting a matrix subroutine from a plurality of subroutines by determining which said matrix subroutine can perform said matrix multiplication consistent with which matrix is selected to reside in said one of said at least one cache.

16. (Original) The signal-bearing medium of claim 14, wherein said matrix subroutine comprises a subroutine from LAPACK (Linear Algebra PACKage).

17. (Currently amended) The signal-bearing medium of claim 16, wherein said ~~LAPACK~~ subroutine comprises a BLAS Level 3 routine or a BLAS Level 3 ~~L1 cache~~ kernel ~~types~~ routine.

18. (Currently amended) The signal-bearing medium of claim 14, wherein data for said second matrix and said third matrix streams from said higher level such that said data from one of said second matrix and said third matrix streams in a vector format ~~and data from the other of said second matrix and said third matrix streams in a scalar format~~ into said L1 cache

19. (Currently amended) A method of providing a service involving at least one of solving and applying a scientific/engineering problem, said method comprising at least one of:

using a linear algebra software package that performs one or more matrix processing operations, said method comprising streaming data for matrices involved in processing said linear algebra subroutines such that data is processed using data for a first matrix stored in a cache as a matrix format and data from a second matrix and a third matrix is stored in a memory device at a higher level than said cache, said streaming providing data from said higher level in a manner as said data is required for said processing by streaming data from two of three matrices involved in processing said linear algebra subroutine to a first matrix such that submatrix data of said two matrices residing in a higher level cache or in a memory is streamed to submatrix data of said first matrix residing in said cache;

providing a consultation for solving a scientific/engineering problem using said linear algebra software package;

transmitting a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result; and

receiving a result of said linear algebra software package on at least one of a network, a signal-bearing medium containing machine-readable data representing said result, and a printed version representing said result.

20. (Currently amended) The method of claim 19, wherein said matrix subroutines comprise BLAS Level 3 ~~L1 cache~~ kernels ~~from a LAPACK(Linear Algebra PACKage)~~ or BLAS Level 3 factorization kernels.